GIT AND GITHUB

TIP: Leave the branches from a remote repo intact, only track them on your local repo. Preferrably, create your own branch for aech feature regarless of the main branch and merge on your "so callled" main/base branch reletive to your work.

GIT AT LOCAL REPO. (Read the docs)

1. Initialise the directory or project using git init, and also set up configurations using git global commands – username and email so git may identify the user.
2. git status - it will show you files staged for commit (green), changes made to files that git is tracked but not yet staged, and untracked files (git is not aware of these)
3. git diff – shows the changes made (for files, commits, directories, etc)
4. git log – shows commits made for a branch: as oneline, as graph – check options
5. git show – starts from the HEAD pointer detailing comits made and who made them.

after cloning a remote repo

6. git will have local repos and remote repos – git branch –all and git remote -v dispalys information about them
7. origin is the name of the remote repo, and origin/main is repo which tracks remote changes.
8. local main must strive to merge to its main counter part – using fetch, merge then push.
9. branches can be merged on a local repo perspective. Also they can be pushed to the remote repo separately and pull request maybe raised to merge it to main. But first the issues must be resolved locally, the branch must be locally ready first by inevstigating main/relevant branch by fetching it first

- git status can be used to insepct different files
- or just mark them as resolved with git add
- git diff –name-status HEAD..origin/main (insepcting changes)

10. You can also decide to contribute directly to main.

11. You can be added as a collaborator and have full access to the repo as the owner – make changes to it and even without raising pull requests.
12. strive to understand the branches separately – their history, HEAD pointers, and have a complete picture of how they differ.
13. Alternatively, you can fork a repository, have three repos your working on

- clone the repo using ssh – set up the necessary keys using ssh-keygen -t -b
- Ensure the forked and the upstream repo are synced – forked repo is not behind. Preferably setting up some remote branches locally
- fetch the changes after syncing the repos or fetch them directly (?)
- commit locally and then raise a pull request.
- Use git checkout -b local remote to create a tracking local branch.
- Use git branch -vv to show which local branch tracks remote
- git push -u local remote – set up tracking first time while pushing
- git branch –set-upstream-to=origin/ - set up after creating local

14.Reset to previous state

- git reset HEAD~1 – to undo the most recent change, current branch
- git reset –hard HEAD-1 – completely undo last commit changes

- git revert <commit-hash> - i think this one is great
- git revert --hard <commit-hash> - i think this one is great and stronger
- git reset --soft HEAD~1 – applied many times per commit